Generating Multiple Hierarchical Segmentations Of Music Sequences Using Adapted Correlative Matrices

Paul Lascabettes, Corentin Guichaoua, Elaine Chew

GENERATING MULTIPLE HIERARCHICAL SEGMENTATIONS OF MUSIC SEQUENCES USING ADAPTED CORRELATIVE MATRICES

Paul Lascabettes, Corentin Guichaoua, Elaine Chew
Sorbonne Université, IRCAM, CNRS, Ministère de la Culture, STMS, Paris, France
{lascabettes,guichaoua,eniale}@ircam.fr

ABSTRACT

Segmentation is an important problem for music analysis, performance, perception, and retrieval. There is often more than one way to segment a piece of music, as reflected in the multiple interpretations of a piece of music. Here, we present an algorithm that can generate multiple hierarchical segmentations of a music sequence based on approximate repeated patterns. A relation between music objects defines this approximation to generate an adapted correlative matrix (ACM). Correlative matrices are data structures for representing repeated patterns that can overlap; ACMs constrain patterns to not overlap. We propose an algorithm that extracts meaningful information from ACMs to identify segmentations in a hierarchical way. Changing the relation produces alternate hierarchical segmentations of the same sequence. The algorithm iteratively selects patterns based on their distinctiveness, i.e. if other patterns begin with the same starting note or immediately after it. We apply this method to various musical objects: a sequence of notes, chords, or bars. In each case, we define different relations on these musical objects and test the method on musical examples to produce multiple hierarchical segmentations. Given a segmentation, the relation that produces that segmentation then gives a possible explanation for that segmentation.

1. INTRODUCTION

Segmentation is an important problem for music analysis, performance, perception, and retrieval. It consists of dividing up a musical sequence into non-overlapping segments. Music segmentation has been studied in the audio and symbolic domain. However, compared to work on audio sources, there has been comparatively less work on segmentation in the symbolic domain [1]. Segmentation tasks with symbolic sources focus on the musical score (i.e. the composition). Lerdahl et al. proposed the *Generative theory of tonal music* (GTTM) [2], where they started to model segmentations in a hierarchical way. This was expanded to include aspects of harmonic tension in *Tonal Pitch Space* [3]. Separate to this, computational approaches based on the GTTM rules were developed [4]. Other algorithms, such as Chew's *Boundary Search Algorithm*, were based on tonality, using key boundaries to create segmentations [5]. More recently, the *Correlative Matrix* was used to first detect and classify patterns, then determine the best segmentation using a score function [6].

Building on these prior work, we develop an algorithm that generates hierarchical segmentations of a musical sequence. Our approach differs from traditional ones in that it provides multiple hierarchical segmentations. These hierarchical segmentations have the potential for explaining different interpretations of the same music which lead to several ways to segment a piece [7, 8]. This kind of approach thus has applications to expressive music performance, and for explaining different perceptions of the same piece.

The multiple hierarchical segmentations generate by our algorithm are based on a chosen approximation. This approximation is defined by a relation between music objects which generates the *Adapted Correlative Matrix* (ACM), a data structure introduced in this paper to represent repeated patterns without overlaps. Our algorithm iteratively selects repeated patterns based on a notion of their distinctiveness, that is to say if other repeated patterns begin at the same time than the starting note (reinforcing the beginning boundary) or immediately after it (reinforcing the end boundary). The distinctiveness criterion exploits the ACM's structure by ensuring that the selected pattern is compatible with other repeated patterns in a hierarchical way. Assuming that the beginning and the end of repeated patterns influence the segmentation of a musical sequence [9], our algorithm iteratively creates boundaries to obtain hierarchical segmentations.

We apply this method to different musical genres and music objects: a sequence of notes, chords and bars. In each case, we define several relations between music objects which yield different hierarchical segmentations. Finally, we visualise the results obtained as a tree and discuss the results. The remainder of this article is structured as follows. Section 2 generalises the existing definition of the Correlative Matrix (2.1) and introduces the Adapted Correlative Matrix (2.2) in order to work with non-overlapping repeating patterns. Section 3 describes the proposed algorithm which extracts meaningful information of the ACM to generate hierarchical segmentations. Section 4 illustrates this method with various music objects, starting with a sequence of notes (4.1), then a sequence of chords (4.2) and also with a sequence of bars (4.3). Finally, Section 5 concludes this paper.

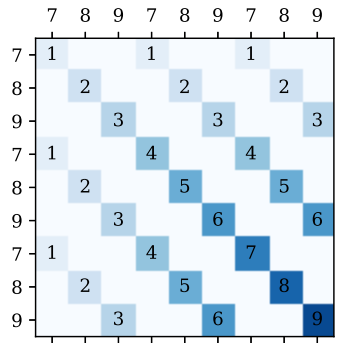


Figure 1. Correlative matrix generated by the sequence $T = (7, 8, 9, 7, 8, 9, 7, 8, 9)$.

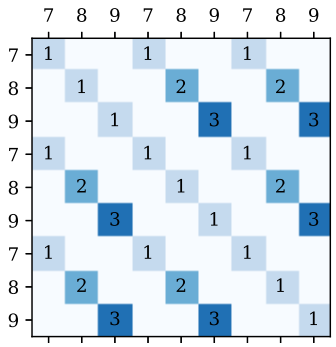


Figure 2. Adapted correlative matrix generated by the sequence $T = (7, 8, 9, 7, 8, 9, 7, 8, 9)$.

2. DEFINITION OF THE ADAPTED CORRELATIVE MATRICES

2.1 Correlative Matrix

The *Correlative Matrix* was first introduced in music processing in [10, 11] in order to detect exact repeating patterns in a sequence of pitches. For a given sequence of pitches $(p_1, \ldots, p_n)$ of length n, the first definition of the correlative matrix was an $n \times n$ matrix, where the coefficient of the i^{th} row and the j^{th} column is set to one if $p_i = p_j$. Moreover, if $p_i = p_j$ and $p_{i+1} = p_{j+1}$, meaning there is one or more repeated patterns of length two, the coefficient of the $i + 1^{th}$ row and the $j + 1^{th}$ column is set to two. By following this process, the value of each coefficient of the correlative matrix indicates the length of a repeating pattern. Compare to the self-similarity matrix used in audio-based music segmentation [12], the correlative matrix allow us to easily detect the longest repeating patterns with the maximal coefficients. Later, the correlative matrix has also been defined to allow for a sequence of intervals or a combination of pitch contours and note durations [6]. Therefore, the coefficient of the correlative matrix was modified to compare if two elements of the sequence were equal (with regard to pitch) [10, 11] or below a similarity threshold (with regard to pitch, contour and duration) [6]. This can be generalised with a symmetric and reflexive relation in order to capture a wide variety of musical objects, indeed the equality or the similarity threshold are both symmetric and reflexive relations. We then generalise the correlative matrix using the following definition:

Definition (Correlative Matrix): Let $T = (t_1, \ldots, t_n)$ be a sequence and $\equiv$ a relation on T which is reflexive ($\forall t_i \in T,\ t_i \equiv t_i$) and symmetric ($\forall t_i, t_j \in T,\ t_i \equiv t_j \Leftrightarrow t_j \equiv t_i$). The *Correlative Matrix* is an $n \times n$ matrix where the coefficient $C_{i,j}$ of the line i and the column j is defined by:

$$C_{i,j} = \begin{cases} C_{i-1,j-1} + 1, & \text{if } t_i \equiv t_j, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

with the convention: $C_{i,j} = 0$ if i or j is negative.

For example, if $T = (7, 8, 9, 7, 8, 9, 7, 8, 9)$ and $\equiv$ is the usual equality $=$ on $\mathbb{R}$ (i.e. $7 \equiv 7$ and $7 \not\equiv 8$), the correlative matrix would be as shown in Figure 1. The following patterns are detected: $(7, 8, 9, 7, 8, 9, 7, 8, 9)$, $(7, 8, 9, 7, 8, 9)$ and $(7, 8, 9)$. The first two patterns are detected in the sequence because overlaps are allowed. In order to use the idea of the correlative matrix to identify contiguous segmentations of T, we need to adapt the definition of the correlative matrix to disallow overlaps. Granted, in music, there exist cases where the last note of a segment can be the first note of the next, but this is outside the scope of this paper.

2.2 Adapted Correlative Matrix

Here, we define the *Adapted Correlative Matrix* where nearly repeated patterns can be easily detected in a sequence T without overlaps.

Definition (Adapted Correlative Matrix): Let $T = (t_1, \ldots, t_n)$ be a sequence and $\equiv$ a relation on T which is reflexive and symmetric. The *Adapted Correlative Matrix* is an $n \times n$ matrix where the coefficient $C_{i,j}$ of the line i and the column j is defined by:

$$C_{i,j} = \begin{cases} C_{i-1,j-1} + 1, & \text{if } t_i \equiv t_j \text{ and} \\ & C_{i-1,j-1} + 1 \leq |i - j|, \\ 1, & \text{if } t_i \equiv t_j \text{ and} \\ & C_{i-1,j-1} + 1 > |i - j|, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

with the convention: $C_{i,j} = 0$ if i or j is negative.

When $t_i \equiv t_j$, the maximal length of the pattern without overlaps is $|i - j|$, as with any longer pattern, the higher index would be in both occurrences of the pattern. Therefore, if $C_{i,j} = |i - j|$ and $t_{i+1} \equiv t_{j+1}$, instead of continuing to increment the value for $C_{i+1,j+1}$ and detecting overlapping patterns, in the adapted correlative matrix we restart at $C_{i+1,j+1} = 1$.

The adapted correlative matrix for the sequence $T = (7, 8, 9, 7, 8, 9, 7, 8, 9)$ and $\equiv$, the equality on T, is presented in Figure 2. The longest detected pattern is now:

$(7, 8, 9)$. Unlike the case with the correlative matrix, the pattern $(7, 8, 9, 7, 8, 9)$ is not detected because this pattern is repeated with overlapping.

The correlative matrix was successfully used to detect nearly repeated patterns in a musical sequence, allowing for overlaps. However, by defining the adapted correlative matrix we can retain the fundamental idea of the correlative matrix while avoiding overlaps, which is required for the music segmentation task.

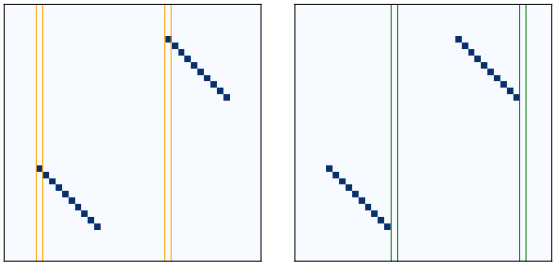
3. ALGORITHM FOR EXTRACTING HIERARCHICAL SEGMENTATIONS USING THE ADAPTED CORRELATIVE MATRIX

Let ACM be the Adapted Correlative Matrix of a sequence $T = (t_1, \ldots, t_n)$ with $\equiv$, a reflexive and symmetric relation. We describe here an algorithm that will extract data from the ACM in order to identify segmentations of T in a hierarchical way.

- **Step 1: Select the longest repeating patterns**
  The *longest repeating patterns* are defined by all the pairs $(t_{i-C_{i,j}}, \ldots, t_i)$ and $(t_{j-C_{i,j}}, \ldots, t_j)$, where $C_{i,j} = \max(\text{ACM})$ (the value of the maximal coefficients of the ACM). Often there will be more than one pattern tied for longest.

- **Step 2: Select the most distinct pair**
  Among all the detected pairs from step 1, the *most distinct one* is the one that maximises the number of repeating patterns that begin at the same time than the starting note (reinforcing the beginning boundary) or immediately after the two patterns of the pair (reinforcing the ending boundary). That is to say, we choose the pair that maximises the number of coefficients equal to 1 in the columns $i - C_{i,j}$ and $j - C_{i,j}$ as illustrated in Figure 3(a) (patterns that start at the same times as the pair) and in the columns $i + 1$ and $j + 1$ illustrated in Figure 3(b) (patterns that start just after the pair).

- **Step 3: Remove the most distinct pairs from the ACM**
  We then update the ACM by removing the most distinct pair and add boundaries to the segmentation at the beginning and end of the most distinct patterns. All coefficients of the most distinct pair become equal to 0 except for the first coefficient which remains equal to 1 (this will be useful in step 2 for future iterations). Moreover, in order to have hierarchical segmentations, if a coefficient $C_{i',j'} > 1$ is on the same column or line as the beginning or the end of a pattern of the most distinct pair, this coefficient will be equal to 1 and $C_{i'+k,j'+k}$ will be $k + 1$ while $C_{i'+k,j'+k} > C_{i'+k-1,j'+k-1}$, which is illustrated in Figure 4(d). With this, a pattern that contains a boundary will be divided in two. This creates a boundary that will remain for the next segmentations and we will thus obtain hierarchical segmentations. Finally, we go back to step 1 until there is no coefficient greater than 1 in the ACM.



(a) Detection of patterns that start at the same time as the pair.

(b) Detection of patterns that start just after the end of the pair.

Figure 3. Criteria for step 2 of the algorithm to choose the most distinct pair of patterns among the longest ones.

Let us take an example with the sequence of real numbers

$$T = (1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 1, 2, 3, 4, 5, 1, 2, 3), \quad (3)$$

with $\equiv$ as the equality on T. The first loop of the algorithm is illustrated in Figure 4. The ACM is represented in Figure 4(a). The maximal coefficient is 8, and two pairs of longest repeating patterns are detected. The pair that maximises the number of patterns that start at the same time and right after the end of the pair is shown in Figure 4(c). Finally, this pair is removed and boundaries are created in Figure 4(d).

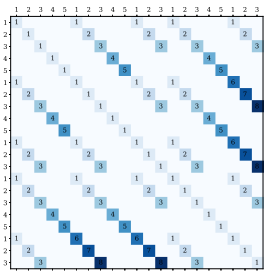
A tree visualisation can be computed to visualize hierarchical segmentations. By representing the length of the detected patterns (the most distinct pairs), the tree visualisation of the previous sequence T (with $\equiv$ the equality) is shown in Figure 5. In this case, our algorithm detects five segmentations that are hierarchically structured. The first segmentation is the sequence itself, so the length of T (here 21) is represented at the top of the tree. The last segmentation is: $(t_1)(t_2)\ldots(t_n)$ where each (t_i) is of length 1, represented by the bottom line $1/1/\ldots/1$ of the tree. The three other detected segmentations of T are:

- (1,2,3,4,5),(1,2,3,4,5,1,2,3),(1,2,3,4,5,1,2,3) represented by the line 5/8/8;

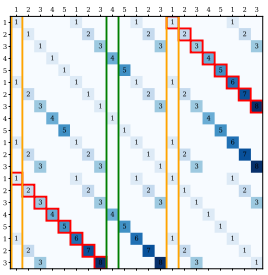
- (1,2,3,4,5),(1,2,3,4,5),(1,2,3),(1,2,3,4,5),(1,2,3) represented by the line 5/5/3/5/3; and,

- (1,2,3),(4,5),(1,2,3),(4,5),(1,2,3),(1,2,3),(4,5),(1,2,3) represented by the line 3/2/3/2/3/3/2/3.

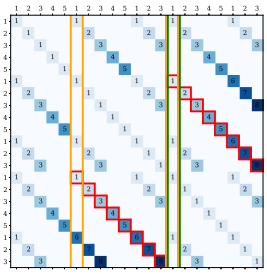
Hierarchical segmentations of musical notes has been studied by Lerdahl and Jackendoff based on different rules on the proximity between notes, repetitions and strong/weak accent of the rhythm in the GTTM [2]. They also represented hierarchical segmentations using a tree visualization. However, the method developed in this paper handles music objects other than notes and is able to propose multiple hierarchical segmentations based on the chosen relation between these objects.



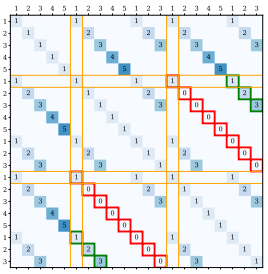
(a) **Step 1:** Two pairs of patterns of length 8 are detected in the ACM.



(b) **Step 2:** For the first pair, 10 patterns start at the same time and 1 pattern starts after the end of the pair (total: 10+1=11).



(c) **Step 2:** For the second pair, 9 patterns start at the same time and 5 patterns start after the end of the pair (total: 9+5=14). This is the most distinct pair.



(d) **Step 3:** The most distinct pair (in red) is removed and boundaries are created at the orange columns and lines (changes in green).

Figure 4. First loop of the algorithm illustrated, these three steps are executed until the coefficients of the ACM are only 0 or 1.

4. APPLICATIONS TO SYMBOLIC MUSIC REPRESENTATIONS

The algorithm presented in Section 3 generates hierarchical segmentations of a sequence with a relation which characterises the chosen approximation for detecting when two patterns are nearly the same. In this section, we apply this algorithm to various symbolic music representations: a sequence of notes (Section 4.1), a sequence of chords (Section 4.2) and a sequence of bars (Section 4.3). In each case, we propose several relations which lead to multiple hierarchical segmentations and we illustrate the results using tree visualisations.

4.1 When T Is a Sequence of Notes

Let T be a sequence of musical notes. There are many ways to define a musical note. For example, a note can be defined as a triplet (p, o, d), as in [13], where p is the pitch, o the onset and d the duration of the note. This can also be enriched to five parameters (o, p, mp, d, v) [14], where mp is the *morphetic pitch* [15] and v the voice where the note occurs. It is also possible to add other parameters such as the velocity, *general pitch interval representation* [16], etc. Among all the different parameters, we first choose here to define a note by its *interval* $\Delta p_i = p_{i+1} - p_i$ (p_i being the pitch of the i^{th} note), then we have $T = (\Delta p_1, \ldots, \Delta p_n)$.

For example, let T be the sequence of intervals of the

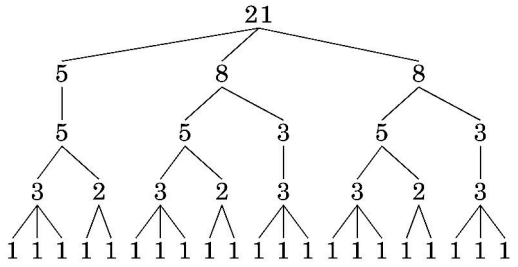


Figure 5. Tree visualisation of the five segmentations in a hierarchical way of the sequence from Equation 3.

first 32 intervals of the Prelude in C major from the *Well-tempered Clavier* by Johann Sebastian Bach (BWV 846). The score is represented in Figure 6. In this case, $T = (4, 3, 5, ..., 5, 3, -18)$. A key advantage of this representation is that it is invariant to transpositions.

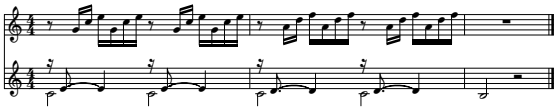


Figure 6. First 32 intervals of J. S. Bach's Prelude in C major, BWV846.

There are also different choices for the relation $\equiv$, because they are many ways to define similarity between intervals [17]. Here we will use a strict interval equality [18], a similarity threshold [6] and an "up/down" relation which defines the melodic contour [19].

Let Δp_i and Δp_j be two intervals of T, we can then define several symmetric and reflexive relations $\equiv$ for T.

- **Strict Intervals Relation:**

$$\Delta p_i \equiv_{st} \Delta p_j \Leftrightarrow \Delta p_i = \Delta p_j. \quad (4)$$

- **Similarity Threshold Intervals Relation:**

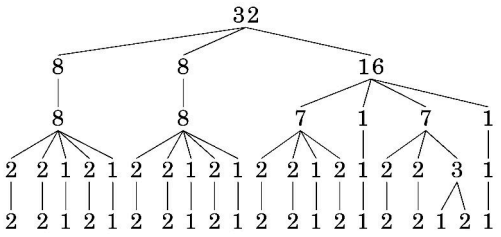
$$\Delta p_i \equiv_{sim} \Delta p_j \Leftrightarrow |\Delta p_i - \Delta p_j| \leq \lambda. \quad (5)$$

- **Melodic Contour Relation:**

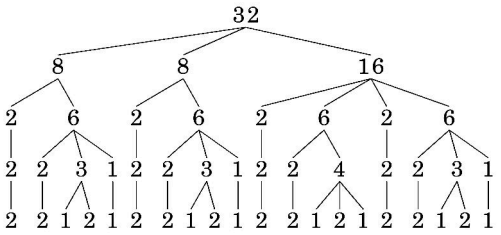
$$\Delta p_i \equiv_{mc} \Delta p_j \Leftrightarrow \mathrm{sgn}(\Delta p_i) = \mathrm{sgn}(\Delta p_j), \quad (6)$$

where sgn is the sign function defined by $\mathrm{sgn}(x) = 1, -1$ or 0 if $x > 0, x < 0$ or $x = 0$.

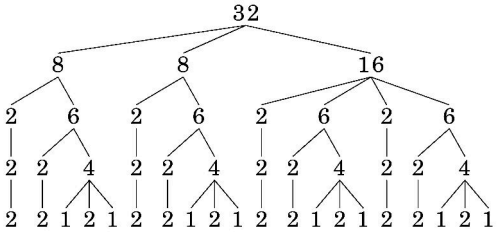
The results of our algorithm for these different relations are illustrated in Figure 7 (the strict interval relation, the similarity threshold intervals relation for $\lambda = 1, 3, 7$ and the melodic contour relation). As before, the different hierarchical segmentations are represented as trees. The ACMs are also represented in Figure 8 for these different relations. It is interesting to note that the less strict the relation is, the more "regular" the tree becomes, until we get the melodic contour relation with very regular and symmetrical segmentations.



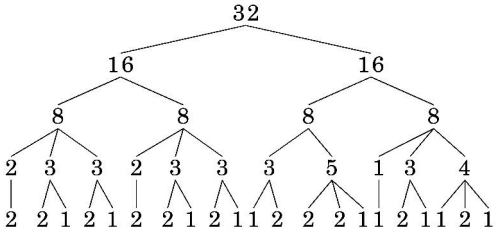
(a) Tree based on the Strict Intervals Relation.



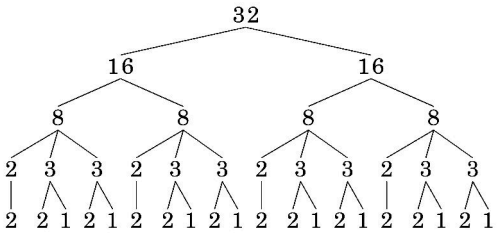
(b) Tree based on the Similarity Threshold Intervals Relation for $\lambda = 1$.



(c) Tree based on the Similarity Threshold Intervals Relation for $\lambda = 3$.

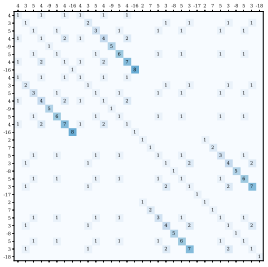


(d) Tree based on the Similarity Threshold Intervals Relation for $\lambda = 7$.

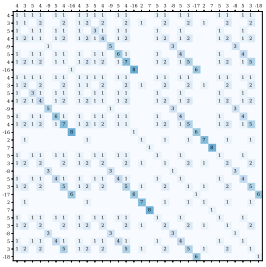


(e) Tree based on the Melodic Contour Relation.

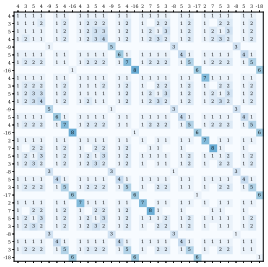
Figure 7. Different trees of the same sequence of notes T (BWV 846) when the relation between intervals changes.



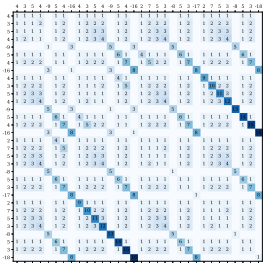
(a) ACM based on the Strict Intervals Relation.



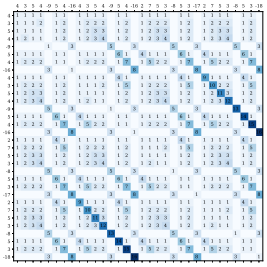
(b) ACM based on the Similarity Threshold Intervals Relation for $\lambda = 1$.



(c) ACM based on the Similarity Threshold Intervals Relation for $\lambda = 3$.



(d) ACM based on the Similarity Threshold Intervals Relation for $\lambda = 7$.



(e) ACM based on the Melodic Contour Relation.

Figure 8. Different ACMs of the same sequence of notes T (BWV 846) when the relation between intervals changes.

In order to work with the durations of the notes, we also applied the algorithm to a sequence of notes defined by $x = (\Delta p, l)$ where l is the length of the note. We defined the relation between two notes $x_i = (\Delta p_i, l_i)$ and $x_j = (\Delta p_j, l_j)$ by:

$$x_i \equiv x_j \Leftrightarrow |\Delta p_i - \Delta p_j| \leq 3 \text{ and } l_i = l_j. \qquad (7)$$

The results of the algorithm is illustrated in Figure 9 with the introduction of Chopin's Mazurka Op.7 No.1. The different colors represent different segmentations. These nearly repeated patterns could also be interesting for performance. For example, the identified patterns could be ones that might be highlighted through prosodic variations like dynamic accents; because the patterns are repeated and distinctive, they could also sound perceptually plausible.

4.2 When T Is a Sequence of Chords

Let $T = (c_1, \ldots, c_n)$ be a sequence of chords where chords of root C are labelled from the set $\hat{\text{C}} = \{\text{C}, \text{C}^6, \text{C}^7, \text{C}^7_M,$

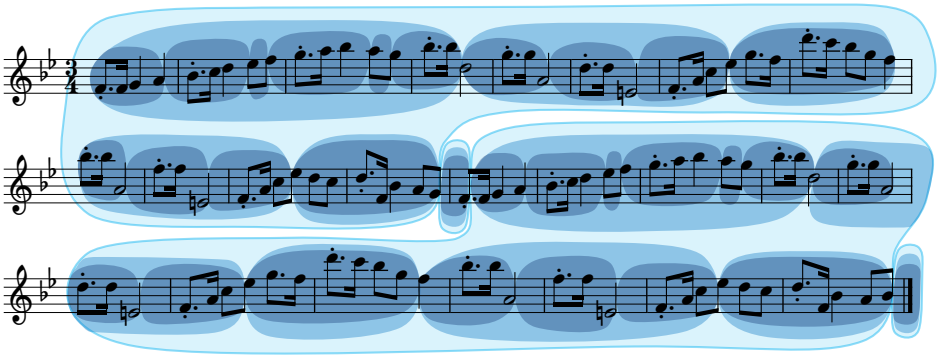


Figure 9. Detected hierarchical segmentations of the melody in Chopin's Mazurka Op.7 No.1 as marked in the score.

C_m, C_m^6, C_m^7, C_m^{M7},C+,C+^{7},C^o,C^{o7},C$^{\emptyset}$} and similarly for other roots. We could define the relation between two chords by strict equality ($c_i \equiv_{st} c_j \Leftrightarrow c_i = c_j$) to have a strong constraint between chords, but this runs the risk of being overly sensitive to small ornamental changes.

Let us instead define the ChordType function of a chord $c_i \in \hat{C}$ (and similarly for other roots) by:

$$\text{ChordType}(c_i) = \begin{cases} C_{maj} & \text{if } c_i = C, C^6, C^7, C_M^7, \\ C_{min} & \text{if } c_i = C_m, C_m^6, C_m^7, C_m^{M7}, \\ C_{aug} & \text{if } c_i = C+, C+^7, \text{ and} \\ C_{dim} & \text{if } c_i = C^o, C^{o7}, C^{\emptyset}. \end{cases} \tag{8}$$

For example $\text{ChordType}(C_m^7) = C_{min}$ or $\text{ChordType}(F\sharp^7) = F\sharp_{maj}$. Let c_i and c_j be two chords of T, we can then define the ChordType Relation $\equiv_{ct}$ on T by:

$$c_i \equiv_{ct} c_j \Leftrightarrow \text{ChordType}(c_i) = \text{ChordType}(c_j). \tag{9}$$

Some other relations, which we will not develop here, could include threshold relations based on the distance of chords within the Tonnetz [20] or one of the parsimonious relations defined by Douthett and Steinbach [21].

Let us take as example the song *In My Life* from The Beatles, released in 1965. All the songs from The Beatles are annotated at `http://isophonics.net` with: structural segmentation, key changes, chords, and beats. According to this database, the chords of the song are: $T =$ (A, E, A, E, A, E, $F_m\sharp$, A^7, D, D_m, A, A, E, $F_m\sharp$, A^7, D, D_m, A, $F_m\sharp$, D, G, A, $F_m\sharp$, B, D_m, A, A, E, A, E, $F_m\sharp$, A^7, D, D_m, A, A, E, $F_m\sharp$, A^7, D, D_m, A, $F_m\sharp$, D, G, A, $F_m\sharp$, B, D_m, A, A, E, $F_m\sharp$, A^7, D, D_m, A, A, E, $F_m\sharp$, A^7, D, D_m, A, $F_m\sharp$, D, G, A, $F_m\sharp$, B, D_m, A, A, E, D_m, A, E, A).

The results of the algorithm applied to this sequence T with the relation $\equiv_{ct}$ defined in (9) are represented in Figure 10. We can compare the hierarchical segmentations obtained by our algorithm with the annotated structure from the database represented in Figure 11.

We can see that the second line (2/2/22/2/22/22/6) contains three main sections of length 22. Each of these sections corresponds to `Verse/Bridge` and the remaining sections of length 2 and 6 map to the `Intro`, `Half-intro` and `Outro`. In the third line, i.e. (2/2/7/7/8/2/7/7/8/7/7/8/6), the `Verse` section is subdivided into two sequences of 7 chords and the `Bridge` is a sequence of 8 chords.

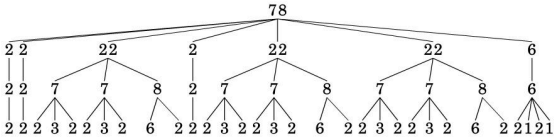


Figure 10. Hierarchical segmentations of the Beatles' *In My Life* based on the chord sequence from `http://isophonics.net`.

```
0.000   0.416   :   silence
0.416   9.616   :   intro
9.616   28.302  :   verse
28.302  46.719  :   bridge
46.719  51.438  :   half-intro
51.438  70.206  :   verse
70.206  88.700  :   bridge
88.700  107.253 :   verse_(instrumental)
107.253 125.659 :   bridge
125.659 143.715 :   outro
143.715 147.973 :   silence
```

Figure 11. Annotated segmentation of *In My Life* from The Beatles.

4.3 When T Is a Sequence of Bars

Let $T = (b_1, \ldots, b_n)$ be a sequence of musical bars. Each bar contains a set of notes (the number of notes can be different from one bar to the next). One way to define a relation $\equiv$ between two bars b_i and b_j is to consider the number of common notes between b_i and b_j. For example, b_i and b_j can be considered in relation if they share at least 50% of their notes, that is to say:

$$b_i \equiv b_j \Leftrightarrow \frac{2|b_i \cap b_j|}{|b_i| + |b_j|} \geq 0.50, \tag{10}$$

where $|b_i|$ is equal to the number of notes of the bar b_i.

With this definition, we compute the hierarchical segmentations of the song as in the previous section i.e.: *In My Life* from the Beatles. The results are represented in Figure 12. The results are similar to Figure 10 with three main sections of equal length for the second line which correspond to the `Verse/Bridge`. The `Intro`, `Half-intro` and `Outro` are also represented on this line. Note that some bars contain more than one chord, e.g. in the `Verse` and `Outro`, which is why there are more chords than bars between Figure 10 and Figure 12.

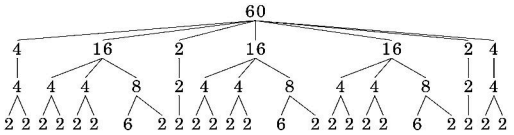


Figure 12. Hierarchical segmentations of *In My Life* from The Beatles based on a sequence of bars.

It is possible to change the relation $\equiv$ between two bars b_i and b_j by adapting the 50% threshold. Also, we can determine the musical chord or key of a bar using a key-finding function, Key, which can be based on the *Krumhansl-Schmuckler Key-Finding Algorithm* [22] or the *Spiral Array Model* [23]. Then, we can define the Key (similar for Chord) Relation $\equiv_{key}$ between two bars b_i and b_j by:

$$b_i \equiv_{key} b_j \;\Leftrightarrow\; \mathrm{Key}(b_i) = \mathrm{Key}(b_j). \tag{11}$$

In this section, we have demonstrated the usefulness of the proposed method with various music objects from symbolic music representations. Several such relations that are musically pertinent have also been presented, but many more can be conceived, including relations that would apply to audio-based objects, for instance, between two frames of a spectrogram. By adjusting which relation to focus on, it is possible to access a much broader meaning of what constitutes a repeated pattern, allowing this approach to be applied to music genres which do not typically exhibit the strong repetitions that are usually required, while preserving the algorithm's lightweight advantage.

5. CONCLUSION

In this paper, we have further expanded the generalisation of *Correlative Matrices* [11] that began with their parametric extension [6] to accept a wide family of relations that define the degree of nearness required to be considered a repeat of a pattern. In order to avoid overlapping patterns, we have introduced the *Adapted Correlative Matrix*, a data structure which represents the repeated patterns of a musical sequence without overlaps. We then defined the novel *distinctiveness* criterion, which characterises the number of repeated patterns that starts at the same time or at the end of a pattern, and proposed an algorithm that iteratively selects patterns based on their distinctiveness to generate hierarchical segmentations.

The representation and algorithms proposed are highly efficient and can scale readily to very large datasets. The examples have shown the versatility of this theory. However, at the moment, human knowledge or feedback are still very important in order to pick a representation and relation that is appropriate for the piece and the desired outcome. We expect that this representation would be a powerful tool for machine learning methods that can tailor relation operators to specific music input and find explanatory models for music segmentation.

Acknowledgments

Paul Lascabettes is funded by a Contrats Doctoraux Spécifiques pour Normalien-nes (CDSN) scholarship. This work is part of the COSMOS project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 788960).

6. REFERENCES

[1] M. Giraud, R. Groult, and F. Levé, "Computational analysis of musical form," in *Computational Music Analysis*. Springer, 2016, pp. 113–136.

[2] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.

[3] F. Lerdahl, "Tonal pitch space," *Music perception*, pp. 315–349, 1988.

[4] D. Temperley, *The cognition of basic musical structures*. MIT press, 2004.

[5] E. Chew, "The spiral array: An algorithm for determining key boundaries," in *International Conference on Music and Artificial Intelligence*. Springer, 2002, pp. 18–31.

[6] B. Rafael and S. M. Oertl, "Mtssm–a framework for multi-track segmentation of symbolic music," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 4, no. 1, pp. 7–13, 2010.

[7] N. Cook, "Performance analysis and chopin's mazurkas," *Musicae scientiae*, vol. 11, no. 2, pp. 183–207, 2007.

[8] P. Lascabettes, C. Agon, M. Andreatta, , and I. Bloch, "Computational Analysis of Musical Structures based on Morphological Filters," in *International Conference on Mathematics and Computation in Music*, Altanta, GA, USA, 2022.

[9] E. Cambouropoulos, "Musical parallelism and melodic segmentation:: A computational approach," *Music Perception*, vol. 23, no. 3, pp. 249–268, 2006.

[10] J.-L. Hsu, A. L. Chen, and C.-C. Liu, "Efficient repeating pattern finding in music databases," in *Proceedings of the seventh international conference on Information and knowledge management*, 1998, pp. 281–288.

[11] J.-L. Hsu, C.-C. Liu, and A. L. Chen, "Discovering nontrivial repeating patterns in music data," *IEEE Transactions on multimedia*, vol. 3, no. 3, pp. 311–325, 2001.

[12] J. Foote, "Visualizing music and audio using self-similarity," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, 1999, pp. 77–80.

[13] M. Giraud, R. Groult, and F. Levé, "Subject and counter-subject detection for analysis of the well-tempered clavier fugues," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2012, pp. 422–438.

[14] D. Meredith, K. Lemström, and G. A. Wiggins, "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," *Journal of New Music Research*, vol. 31, no. 4, pp. 321–345, 2002.

[15] D. Meredith, "The computational representation of octave equivalence in the western staff notation system," in *In Cambridge Music Processing Colloquium*. Citeseer, 1999.

[16] E. Cambouropoulos, "A general pitch interval representation: Theory and applications," *Journal of New Music Research*, vol. 25, no. 3, pp. 231–251, 1996.

[17] M. Giraud, R. Groult, E. Leguy, and F. Levé, "Computational fugue analysis," *Computer Music Journal*, vol. 39, no. 2, pp. 77–96, 2015.

[18] L. Smith and R. Medina, "Discovering themes by exact pattern matching," in *Proceedings of the International Conference on Music Information Retrieval*. Citeseer, 2001.

[19] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: Musical information retrieval in an audio database," in *Proceedings of the third ACM international conference on Multimedia*, 1995, pp. 231–236.

[20] C. L. Krumhansl, "Perceived triad distance: Evidence supporting the psychological reality of neo-riemannian transformations," *Journal of Music Theory*, vol. 42, no. 2, pp. 265–281, 1998.

[21] J. Douthett and P. Steinbach, "Parsimonious graphs: A study in parsimony, contextual transformations, and modes of limited transposition," *Journal of Music Theory*, pp. 241–263, 1998.

[22] D. Temperley, "What's key for key? the krumhansl-schmuckler key-finding algorithm reconsidered," *Music Perception*, vol. 17, no. 1, pp. 65–100, 1999.

[23] E. Chew, "Modeling tonality: Applications to music cognition," in *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, 2001, pp. 206–211.